# 一种利用情感词统计信息构造文本特征表示的方法 *

韩彤晖，杨东强，马宏伟

(山东建筑大学 计算机科学与技术学院，济南 250100)

**摘　要：**数据表达方法和文本分类的效果密切相关。文本分类中常用的数据表达方法主要包括基于词典的共现频率方法、基于隐性语义空间(LSA/SVD)的方法、基于神经网络语言模型的方法。提出一种利用单词的统计特征创建文本分类中特征空间的表达方法。该方法利用单词的七种常见的统计特征，通过相关性分析选取相对独立的统计特征创建特征空间。该方法能够有效降低文本向量空间的维度，同时降低了语义空间内的计算复杂度。情感分类实验的结果表明，与现有的单词的数据表达方法相比，该方法能够显著提高分类算法的准确率和召回率。

**关键词：**数据表达；统计特征；情感分类

**中图分类号：**TP391.1　　doi: 10.3969/j.issn.1001-3695.2018.01.0035

## Novel method of using statistical information to construct feature representation in sentiment classification

Han Tonghui, Yang Dongqiang, Ma Hongwei

(*School of Computer Science & Technology Shandong Jianzhu University*, *Jinan* 250100, *China*)

**Abstract:** Data representation is closely related to the performance of text classification method. There exist three typical methods, namely lexical co-occurrence, Latent Semantic Analysis (LSA) or Latent Semantic Analysis (LSA) or Singular value decomposition (SVD) , and various neural language models. This paper introduces a feature space construction method only using statistical information. The method first collects 7 types of common word's statistical information, and then chooses independent features through correlation analysis, to contrast word feature space vector. This method can effectively reduce the dimension size of vector space models, and can effectively lower computation complexity in deriving latent semantic space. The sentiment classification results shows that in contrast with those current data representation methods, our method can significantly improve the accuracy and recall rates for different classifier.

**Key words:** data representation; statistical features; sentiment classification



## 0　引言

文本的数据表达是文本分类研究的基础，目的是将单词转换为可以被计算机处理的形式。文本的数据表达方式的好坏直接影响到分类算法的准确率。目前文本的数据表达方法主要包括基于词典的共现频率方法[1]、基于隐性语义空间(LSA/SVD)的方法[2]、基于神经网络语言模型的方法[3]。基于词典的共现频率方法将单词在词典内的位置信息和单词在文本内的分布频率作为文本数据表达的基础，该方法简单易行，但是该方法处理大规模数据时会生成稀疏、高维的文本矩阵，降低分类算法的效率。基于隐性语义空间模型的方法通过词-文档矩阵描述单词在文本中的分布信息，通过压缩得到上述矩阵的低阶近似矩阵，通过该近似矩阵实现文本的数据表达，该方法适用于文本主题抽取[2]、单词聚类[4]等研究中。与基于词典的共现频率方法相似该算法执行时需要创建高维稀疏矩阵。矩阵分解过程中奇异值计算的时间复杂度为 O($n^3$)，$n$ 表示数据规模，并且压缩操作使生成的低阶矩阵解释性较差。基于神经网络语言模型的方法以单词的分布信息为依据，通过多层神经网络得到词向量，这类方法被广泛应用于词义相似性计算[5]，与基于词典的方法和基于隐性语义空间的方法相比，深度学习的方法具有更好的可扩展性。但是，相较于基于词典的共现频率方法，通过深度学习方法生成的词向量每一维的特征是难以解释的，因此难以对词向量做进一步分析。Kim、Zhang[6, 7]等人利用神经网络直接产生文本的特征表示，本文主要研究单词表达方法对文本情感分类的影响，因此没有将利用深度学习的方法直接产生文本特征并进行分类的方法列入本文的比较范围。

本文提出一种通过组合常见的统计特征，实现单词和文本的数据表达的方法。统计特征根据单词在文本内的分布规律，

**收稿日期：**2018-01-19；**修回日期：**2018-03-14　**基金项目：**国家社科基金资助项目（17BYY19）；国家教育部人文社科基金资助项目（15YJA740054）

**作者简介：**韩彤晖（1991-），男，山东滨州人，硕士研究生，主要研究方向为自然语言处理（kevinsdj@sina.com）；杨东强（1970-），男，副教授，博士，主要研究方向为人工智能；马宏伟（1969-），男，教授，博士，主要研究方向为计算机网络应用.

反映该单词影响文本类别划分的强度。在文本分类研究中，统计特征是提取关键词的参照标注[8]。基于统计特征的关键词抽取方法利用单词的特征值表示单词，根据特征值判断单词能否成为关键词。利用统计特征实现单词的数据表达方法首先选取七种常见的统计特征，再通过相关性分析得到相对独立的统计特征，利用这些独立的统计特征实现单词和文本的数据表达。该方法有效降低了文本空间向量的维度，具有隐性语义空间(LSA/SVD)的压缩效果，与隐性语义空间相比，该方法的时间复杂度仅为 O($n$)。文本分类的实验结果表明，与基于词典的共现频率的方法和基于神经网络语言模型的方法相比，通过单词的统计特征创建文本分特征空间的方法使支持向量机、决策树、随机森林的分类结果具有更高的准确率。

## 1　相关研究

One-hot-vector 一种典型的基于词典的共现频率的表达方法[9]，该方法操作简单，适用于处理小规模数据集，但是当处理大规模的数据时，该方法会占用大量资源，效率也随之降低。常见的基于神经网络语言模型的方法包括基于卷积神经网络(CNN)的方法、基于 Word2Vec[10~12]模型的方法、基于 GloVe[13]模型的方法等。基于卷积神经网络的方法将训练集中的单词作为基本单元，以 one-hot-vector 的形式表示单词，经过多层卷积和池化操作得到最终的单词数据表达形式，该方法的效果取决于神经网络的层数以及训练集的容量，适用于处理大规模语料库，程序执行过程会占用大量计算资源。Weston[14]创建了一个多层 CNN 模型，实现半监督方式对单词进行数据表达，并证明输出层采用的激活函数也会影响实验结果。Meng[15]对目标单词的信息源进行集中处理，将处理的信息引入 CNN 创建词向量，该方法在机器翻译中取得较好的效果。使用 Word2Vec 创建单词的数据表达的方法得到了广泛的关注，首先建立一个词汇表，词汇表内部的单词的表达形式与基于 CNN 的方法类似。之后将原始的词向量送入隐藏层，利用单词的分布信息生成最终的词向量。Word2Vec 的数据表达方法能够根据被分析文本所属的领域选取合适的语料库训练模型，具有良好的扩展性，该方法所生成的词向量的解释性较差。Segura-Bedmar 等人[16]将 Wikipedia 和 MedLine 作为训练语料库，用 Word2Vec 得到词向量，使系统能够在生物医学类文本中识别药品名称。Yang 等人[17]通过使用 Word2Vec 以 Wikipedia 中文语料库为训练集构造模型，实现中文单词的数据表达。Google News[1]模型是基于 Word2Vec 创建的语言模型，将谷歌新闻数据集作为训练语料库，该模型包含三百万个单词，生成词向量的维度为 300。Ghosal[18]通过 Google News 模型生成词向量，利用生成的词向量构造文本矩阵，将文本矩阵送入 CNN-LSTM 网络，实现文本情感分类。Glove 是一种无监督的矩阵生成方法，通过语料库中单词之间的共现频率建立语言模型，最终实现单词的数据表达。使用 GloVe 模型时需要创建单词的共现矩阵，因此该模型处理大规模的数据时会占用大量的计算资源。Lee 等人[19]将 Word2Vec 和 GloVe 模型同 WordNet 词典结合提出一种单词编码方法，该方法能够有效保留单词的语义信息，适用于语义相似性研究，但是这种方法难以有效处理大规模数据。

统计特征能够以数值的形式反映单词的分布信息，在关键词提取实验中特征值可以作为衡量单词与文本类型之间关联强度的指标。Rajeswari[20]将信息增益作为衡量单词与文本类型关联强度的指标。Uysal[21]结合信息增益和让步比，实现从文本中提取关键词。Jana[22]用单词与文本类型之间的共现频率计算互信息，根据单词的互信息取值过滤非关键词。Mesleh[23]将卡方统计量作为单词权重，并以此作为标准，判断单词是否影响文本的类型。Mitra[24]使用相关系数作为衡量单词与文本之间关联程度的标准。Habibi[25]根据候选单词在不同主题的文本内分布频率的差异判断该单词是否为关键词。在文本集合中，关键词与统计特征之间近似一对一的关系。根据上述原因，本文选取 7 种常见的统计特征，通过相关性分析，使用相对独立的特征作为特征空间的元素，实现文本数据表达。

## 2　统计特征与文本的数据表达

### 2.1　统计特征与特征值计算

根据文献[26]中关于单词统计特征的描述，选取七种分布统计特征创建单词数据表达。以下是这七种统计特征的计算公式。

#### 2.1.1 统计特征

1)信息增益(information gain，IG)

信息增益的计算公式描述如下：

$$IG(w)=S(w)\times\left\{-\sum_{C\in\Omega}P(C)\times\log P(C)\right\}-\left\{\sum_{t\in\{w,\bar{w}\}}P(t)\times\left[-\sum_{C\in\Omega}P(C|t)\times\log P(C|t)\right]\right\} \tag{1}$$

该公式在原始信息增益的基础上乘以单词的情感值 $S(w)$，$S(w)$取值为-1、1，使 IG 可以映射单词的情感极性。

2)让步比(odds ratio，OR)

让步比的计算公式如下：

$$OR(w,C)=\log\frac{P(w|C)\times\left[1-P(w|\bar{C})\right]}{\left[1-P(w|C)\right]\times P(w|\bar{C})} \tag{2}$$

3)互信息(mutual information，MI)

单词互信息的计算公式如下：

$$MI(w,C)=\log\frac{P(w|C)}{P(w)} \tag{3}$$

4)对数概率比(log probability ratio，LPR)

以下是单词对数概率比的计算公式：

$$LPR(w,\ C)=\log\frac{P(w|C)}{P(w|\bar{C})} \tag{4}$$

1 https://github.com/mmihaltz/word2vec-GoogleNews-vectors

5)卡方统计(Chi-squire，Chi)

单词卡方值的计算公式如下所述：

$$\chi^2(w,C)=\frac{(A\times D-E\times B)^2}{(A+E)\times(B+D)\times(A+B)\times(E+D)} \tag{5}$$

6)相关系数(correlation coefficient，CC)

相关系数的计算公式如下所述：

$$CC(w,C)=\frac{\left[P(w,C)P(\overline{w},\overline{C})-P(w,\overline{C})P(\overline{w},C)\right]}{\sqrt{P(w)P(\overline{W})P(C)P(\overline{C})}} \tag{6}$$

7)差异性分布(differential distribution，DD)

差异性分布的公式描述如下：

$$DD(w)=S(w)\times P(w)\times\frac{sd(w)}{\max\limits_{C\in\Omega}\{P(C|w)\}} \tag{7}$$

$sd(w)$表示上述条件概率的标准差，$S(w)$的意义与 IG 部分的介绍相同，即差异性分布公式需要在原公式的基础上乘以单词情感值。

2.1.2 特征值计算

使用元组 $Q_W=\langle q_{C1},\overline{q_{C1}},\cdots,q_{Ci},\overline{q_{Ci}},\cdots,q_{Cn},\overline{q_{Cn}}\rangle$，记录单词 $w$ 在不同情感极性的文本集合中的分布频率。$q_{Ci}$表示包含 $w$，且情感极性为 $Ci$ 的文本的频率，$\overline{q_{Ci}}$ 表示不包含 $w$，且情感极性为 $Ci$ 的文本的频率。上述统计特征的计算方法基本相似，以计算'*fake*'在 IMDB 文本集内部的信息增益和让步比为例，描述特征值计算的实现过程。

IMDB 分为积极性评论和消极性评论，$Q_{fake}$ 的格式为 $Q_{fake}$=<113,12387,334,12166>，即包含和不包含'*fake*'的积极性文本的频率分别为 113、12387；包含和不包含'*fake*'的消极性文本的频率分别为 334、12166。通过 $Q_{fake}$ 得到计算'*fake*'的信息增益和让步比所需的概率，结果如表 1 所示。

表 1　概率计算结果

| 类型 | 取值 | 类型 | 取值 |
|---|---|---|---|
| $P(fake)$ | 0.0179 | $P(positive\|\overline{fake})$ | 0.5045 |
| $P(\overline{fake})$ | 0.9821 | $P(negative\|\overline{fake})$ | 0.4955 |
| $P(positive)$ | 0.5000 | $P(fake\|positive)$ | 0.0090 |
| $P(negative)$ | 0.5000 | $P(fake\|\overline{positive})$ | 0.9910 |
| $P(positive\|fake)$ | 0.2528 | $P(fake\|negative)$ | 0.0267 |
| $P(negative\|fake)$ | 0.7472 | $P(fake\|\overline{negative})$ | 0.9733 |

在情感词典中，'*fake*'被标注为消极性情感词，因此 $S$(fake)=-1。经过计算得到'*fake*'的信息增益：IG(fake)=-0.2324，'fake'在积极性文本中的让步比：OR(fake,*positive*)=-1.1018，在消极性文本中的让步比：OR(fake,*negative*)=1.1.18。

2.2　单词数据表达

通过特征值创建词向量。其中，图 1 展示了创建词向量的具体流程。

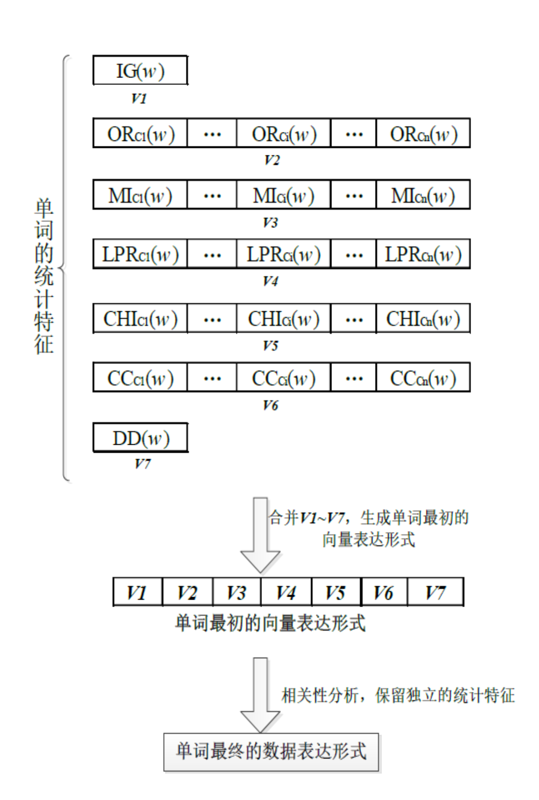



图 1　词向量的创建流程

经过计算得到单词的上述 7 种特征值，利用得到的特征值创建特征值向量 ***V1~V7***。根据在 3.1 中描述的统计特征的计算公式可知，IG 与 DD 反映单词在语料库中整体的统计特征，向量 ***V1*** 和 ***V7*** 的维度为 1，而 OR~CC 记录单词在不同极性文本内的统计特征，因此，向量 ***V2~V6*** 的维度为$|\Omega|$，其中，$|\Omega|$表示文本集中包含$|\Omega|$类文本。合并 ***V1~V7*** 构造词向量 $V_W$，维度为 $5|\Omega|+2$。使用情感词典中的所有词向量构造矩阵，分析单词的各种统计特征之间的相关性，保留相对独立的特征。

以'fake'对应的词向量 $V^T_{fake}$ 的创建过程为例，表 2 展示了'fake'在文本集 IMDB 中的统计特征。

表 2　'*fake*'在 IMDB 内的统计特征

| 统计特征 | 取值 | 统计特征 | 取值 |
|---|---|---|---|
| IG(*fake*) | -0.2324 | LPR(*fake,negative*) | 1.0838 |
| OR(*fake,positive*) | -1.1018 | CHI(*fake,positive*) | 0.0323 |
| OR(*fake,negative*) | 1.1018 | CHI(*fake,negative*) | 0.0323 |
| MI(*fake,positive*) | -0.0122 | CC(*fake,positive*) | -10.5477 |
| MI(*fake,negative*) | 0.0072 | CC(*fake,negative*) | 10.5477 |
| LPR(*fake,positive*) | -1.0838 | DD(*fake*) | -0.0183 |

利用特征值创建向量 V1 $^T$ ~V7 $^T$，得到'fake'最初的数据表达形式 $V'^T$，格式如下：

$$V'^T=\begin{pmatrix}-0.2324,-1.1018,1.1018,-0.0122,0.0072,-1.0838,\\1.0838,0.0323,0.0323,-10.5477,10.5477,-0.0183\end{pmatrix}$$

分析词向量内部各个特征之间的相关性，选取相对独立的统计特征。表 3 为上单词的统计特征在 IMDB 语料库中的相关

系数。

表 3　IMDB 内统计特征之间的相关系数

| | IG | $OR_{pos}$ | $OR_{neg}$ | $MI_{pos}$ | $MI_{neg}$ | $LPR_{pos}$ | $LPR_{neg}$ | $CHI_{pos}$ | $CHI_{neg}$ | $CC_{pos}$ | $CC_{neg}$ | DD |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IG | 1 | -0.11 | 0.11 | -0.59 | 0.18 | -0.10 | 0.10 | 0.82 | 0.82 | -0.25 | 0.25 | -0.32 |
| $OR_{pos}$ | -0.11 | 1 | -1 | 0.41 | -0.43 | 0.99 | -0.99 | -0.05 | -0.05 | 0.80 | -0.80 | 0.40 |
| $OR_{neg}$ | 0.11 | -1 | 1 | -0.41 | 0.43 | -0.99 | 0.99 | 0.05 | 0.05 | -0.80 | 0.80 | -0.40 |
| $MI_{pos}$ | -0.59 | 0.41 | -0.41 | 1 | -0.89 | 0.40 | -0.40 | -0.53 | -0.53 | 0.80 | -0.80 | 0.93 |
| $MI_{neg}$ | 0.18 | -0.43 | 0.43 | -0.89 | 1 | -0.42 | 0.42 | 0.26 | 0.26 | -0.84 | 0.84 | -0.97 |
| $LPR_{pos}$ | -0.10 | 0.99 | -0.99 | 0.40 | -0.42 | 1 | -1 | -0.04 | -0.04 | 0.79 | -0.79 | 0.38 |
| $LPR_{neg}$ | 0.10 | -0.99 | 0.99 | -0.40 | 0.42 | -1 | 1 | 0.04 | 0.04 | -0.79 | 0.79 | -0.38 |
| $CHI_{pos}$ | 0.82 | -0.05 | 0.05 | -0.53 | 0.26 | -0.04 | 0.04 | 1 | 1 | -0.20 | 0.20 | -0.36 |
| $CHI_{neg}$ | 0.82 | -0.05 | 0.05 | -0.53 | 0.26 | -0.04 | 0.04 | 1 | 1 | -0.20 | 0.20 | -0.36 |
| $CC_{pos}$ | -0.25 | 0.80 | -0.80 | 0.80 | -0.84 | 0.79 | -0.79 | -0.20 | -0.20 | 1 | -1 | 0.81 |
| $CC_{neg}$ | 0.25 | -0.80 | 0.80 | -0.80 | 0.84 | -0.79 | 0.79 | 0.20 | 0.20 | -1 | 1 | -0.81 |
| DD | -0.32 | 0.40 | -0.40 | 0.93 | -0.97 | 0.38 | -0.38 | -0.36 | -0.36 | 0.81 | -0.81 | 1 |

根据统计特征之间的相关系数，选取相对独立的统计特征。实验结果表明，在 IMDB 中，将相关系数绝对值小于 0.85 的两个统计特征认定为相对独立时，系统效率最高。最终得到'fake'的词向量 $V_{fake}^{T}$，格式如下：

$$V_{fake}^{T}=(-0.2324,-1.1018,-0.0122,10.0323,-10.5477)$$

### 2.3　文本的数据表达

使用基于词袋模型(bag of words)的方法构造文本的空间向量模型。构造文本向量的表达式如下：

$$V_T=\sum_{u=1}^{|lexicon|} signal(w_u)\times V_w^u \tag{8}$$

其中：$|lexicon|$表示情感词典内单词的数量，$u$ 表示单词在词典中的编号，$signal(w_u)$为符号函数，指示 $w_u$ 是否在文本 $T$ 中出现，若 $w_u\in T$ 则 $signal(w_u)=1$，否则 $signal(w_u)=0$。

将语料库中的文本以列表的形式存储，并将列表命名为 $L_T$，读入情感词典 $L$。从 $L_T$ 的表头位置读取文本 $T$，创建文本空间向量 $\boldsymbol{V_T}$，并将该向量初始化为 0 向量；遍历 $L$，若单词 $w$ 属于 $T$ 和 $L$ 的交集，则 $\boldsymbol{V_T}=\boldsymbol{V_T}+\boldsymbol{V_w}$。

以创建文本 $T$ 的空间向量模型为例，介绍文本向量的构造过程。$T$ 选自 IMDB，具体表述如下：

"**Great** movie and the family will **love** it! If kid be bore one day just pop the tape in and you will be so **glad** you do!"

遍历 $L$ 得到，$T\cap L$={'*great*','*love*','*glad*'}。情感词'*great*'、'*love*'和'*glad*'的数据表达格式如下：

$$V_{great}^{T}=(2.1278,0.9512,0.0783,4.5879,32.3725)$$

$$V_{love}^{T}=(1.3181,0.8848,0.0586,2.3426,25.5123)$$

$$V_{glad}^{T}=(0.0300,0.3760,0.0029,0.0042,3.8565)$$

该文本的空间向量 $\boldsymbol{V_T}=\boldsymbol{V_{great}}+\boldsymbol{V_{love}}+\boldsymbol{V_{glad}}$，最终的计算结果如下：

$$V_{T}^{T}=(3.4759,2.2120,0.1398,6.9347,61.7413)$$

## 3　文本情感分类测试

本文采用基于统计特征的单词数据表达方法、one-hot-vector 方法、基于词频的方法、基于词频-压缩的方法、基于 CNN 创建单词数据表达的方法，得到情感词的向量模型，并通过 Word2Vec 提供的 Google News 模型直接得到情感词向量，通过对测试文本进行情感分类，分析以上各种单词数据表达方法对分类算法的影响。其中，one-hot-vector 方法和基于单词频率的方法为传统的基于词典的共现频率单词的方法，基于 CNN 的单词数据表达方法属于基于神经网络语言模型的方法。实验采用的文本集分别为 IMDB、yelp2013、yelp2014，情感词典采用 MPQA[2]。上述文本集的基本信息如表 4 所示。

表 4　文本集合的基本信息

| | 文本集 | 分类 | 文本总量 | 各类情感极性的文本的数量 | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | high-positive | positive | neutral | negative | high-negative |
| 训练集 | IMDB | 2 | 25000 | — | 12500 | — | 12500 | — |
| | yelp2013 | 5 | 62522 | 17167 | 26057 | 11989 | 5130 | 2179 |
| | yelp2014 | 5 | 183019 | 50312 | 72740 | 36346 | 16218 | 7231 |
| 测试集 | IMDB | 2 | 25000 | — | 12500 | — | 12500 | — |
| | yelp2013 | 5 | 8671 | 2447 | 3567 | 1607 | 751 | 299 |
| | yelp2014 | 5 | 25399 | 7059 | 9946 | 5118 | 2221 | 1055 |

图 2 展示了实验的基本流程，该实验包含 3 个阶段，第一阶段包含三个步骤：文本预处理、统计单词的分布频率、特征值计算；第二阶段同样包含三个步骤：创建词向量、相关性分析、创建文本向量空间模型；第三阶段的任务为文本情感分类。

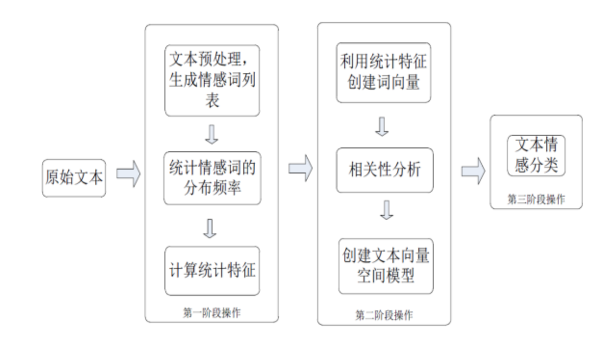



图 2　实验流程图

该文将在 3.1 中对文本预处理操作进行详细介绍；第一阶段第 2 步操作统计情感词在各种情感极性的文本内的分布频率，并将频率信息存入元组 $Q$，例如'*fake*'在 IMDB 内的分布信息为 $Q_{fake}$=<113,12387,334,12166>，根据元组记录的信息计算情感词的 7 种统计特征，其中计算方法如 2.1.2 所。依据 2.2 的描述执行第二阶段的前两步操作，即创建词向量、相关性分析，创建词向量最初的表达形式，再通过相关性分析保留词向量内部相对独立的统计特征，最后得到词向量的最终形式；根据 2.3 的描述，创建文本向量空间模型，实现文本的数据表达。最后执行第 7 步，将生成的文本向量送入分类器，得到分类结果。

### 3.1　文本预处理

实验之前对上述文本集合进行如下操作：

a)规范化(normalization)。将文本中出现的大写字母转换为

2 http://mpqa.cs.pitt.edu/lexicons/

小写字母，去除文本中包含的特殊符号，使用语义相近的单词替代文本中出现的表情符号（例如：':-D '→'happy'、':-(' →'sad'）。

b)词干处理(stemming)。还原文本中出现的名词复数、形容词比较级、动词第三人称单数形式、动词过去式等形式的单词。

c)情感词抽取。被抽取的单词在文本集合中出现的频率大于 $\beta$(根据上述 3 个文本集合内单词总量，分别将 $\beta$ 设置为 15、35、55)，并且单词属于词典 MPQA。

表 5 展示了预处理操作结束后各文本集合内部的单词总量、单词种类以及情感词种类的统计结果。

表 5 文本集合内部的单词信息

| 语料库 | 单词总量 | 单词种类 | 情感词种类 |
|---|---|---|---|
| IMDB | 6025237 | 7666 | 2808 |
| yelp2013 | 9901789 | 54483 | 1914 |
| yelp2014 | 30087844 | 98566 | 2660 |

### 3.2 创建词向量和文本向量

基于统计特征的单词的数据表达方法根据单词在文本集中的七种特征值创建情感词向量，方法如 3.2 所述。经过相关性分析后得到在 IMDB 中词向量的维度为 5，在 yelp2013 和 yelp2014 中词向量的维度均为 8。实验表明在 IMDB 中最佳的统计特征组合为：IG、$OR_{pos}$、$MI_{pos}$、$CHI_{pos}$、$CC_{pos}$；在 yelp2013 中最佳的统计特征组合为：IG、$OR_{high\text{-}pos}$、$OR_{neutral}$、$MI_{neg}$、$CHI_{high\text{-}pos}$、$CHI_{neutral}$、$CC_{high\text{-}pos}$、$CC_{high\text{-}neg}$；在 yelp2014 中最佳的统计特征组合为：IG、$OR_{high\text{-}pos}$、$OR_{neutral}$、$MI_{neg}$、$CHI_{high\text{-}pos}$、$CHI_{neutral}$、$CC_{high\text{-}pos}$、$CC_{pos}$。

对比实验中情感词的数据表达方法分别为：one-hot-vector 方法、基于单词频率的方法、基于 CNN 的单词数据表达方法、Word2Vec 模型，描述如下：

a)one-hot-vector 方法。词向量由 0、1 组成，向量长度等于情感词典内单词的数量。例如情感词为 $w_u$，u 表示该单词在情感词典中的序号，则该单词对应的 one-hot-vector 向量的第 u 位元素为 1，其他位置的元素均为 0。

b)基于词频的方法。该向量的创建方法与 one-hot-vector 方法相似，向量长度等于情感词典内的单词数量，在情感词 $w_u$ 对应的词向量中，第 u 位元素为 $w_u$ 的 TF-IDF 值，其余元素均为 0。

c)基于词频-压缩的方法。首先使用基于词频的方法创建文本空间模型，再通过 SVD 分解矩阵得到奇异值，根据奇异值设置压缩后文本空间的，经过实验可知，当 IMDB、yelp2013、yelp2014 的文本空间维度分别设置为 75、125、130 时，分类器具有最佳效果。

d)基于 CNN 创建单词数据表达的方法。分别将 IMDB、yelp2013、yelp2014 作为语料库，CNN 根据情感词在对应语料库中的分布信息生成单词的数据表达模型。生成词向量的维度分别设置为 5、10、20、30、40、50、60，实验证明向量维度为 40 时分类效率达到最优。

e)Word2Vec 模型。通过 Word2Vec 对谷歌新闻语料库(大约包含 1000 亿个单词)进行训练得到的单词的数据表达模型。该模型包含三百万个词向量，由于模型设置的缺省向量维度为 300，因此实验不再修改向量维度。

实验中采用基于词袋模型的方法创建文本向量空间模型，向量的创建方法与图 2 中步骤 6 的描述相同。

### 3.3 文本情感分类

实验依次选择支持向量机(SVM)、朴素贝叶斯(naïve Bayes)、决策树(decision tree) 、随机森林(random forest)作为分类算法，用于测试上述文本的数据表示方法对分类算法效率的影响程度。其中，实验使用数据处理工具 Weka 提供的分类器，分类器的参数为 Weka 设置的缺省值。

SVM 分类算法的核心观点是通过运算找到一个超平面，利用该平面能够将具有某一类特征的数据从整个数据集中分离。实验表明 SVM 算法在文本情感分类研究中具有较高的效率[27]。

朴素贝叶斯是一种简单的分类算法，该算法求解给定的分类项出现的条件下各个类别出现的概率，将分类项归属于出现概率最大的类别。朴素贝叶斯分类器被普遍应用于文本分类[28]和垃圾邮件过滤[29]。

决策树又称判定树，是一种树型结构，分支节点表示对某一属性的一次检测，每条边为对应的测试结果，叶节点表示类别标记。决策树的执行过程从根节点开始，待分类项与中间节点中的属性进行比较，根据比较结果选择对应的分支，直到叶节点确定待分类项的类别。决策树算法在文本特征提取中具有较高的效率[30]。

随机森林是一种将多棵树集成为一体的学习算法，该算法的基本单元为决策树。对于待分类项，多棵决策树会有多种不同的投票结果，随机森林将待分类项划分到投票次数最多的类型中。Buscaldi[31]证明随机森林在语义相似性计算中的效率优于其他分类器。

## 4 实验结果及分析

### 4.1 实验结果展示

表 6 展示了文本情感分类的结，加粗字体表示分类器的准确率能够达到的最大值。实验结果表明，与其他的数据表达方法相比，基于统计特征创建文本数据表达的方法使分类算法具有更高的准确率。当采用统计特征进行文本数据表达时，使用 SVM 分类器对 IMDB、yelp2013、yelp204 进行情感分类的准确率依次为：84.2%、50.4%、48.1%；使用朴素贝叶斯分类器对上述文本集进行分类的准确率分别为：81.0%、39.6%、39.1%；使用决策树分类器的结果分别为：83.9%、42.6%、40.7%；使用随机森林的分类结果依次为 84.2%、49.3%、47.6。SVM 在文本情感分类测试中具有最高的准确率，朴素贝叶斯算法更适合对基于词典的共现频率方法创建文本向量进行分类测试，随机森林的分类效果优于决策树的分类效果。

表 6　文本分类结果

| | Method | IMDB | | yelp2013 | | yelp2014 | |
|---|---|---|---|---|---|---|---|
| | | P | R | P | R | P | R |
| SVM | one-hot-vector | 77.6% | 77.6% | 49.2% | 47.6% | 46.1% | 45.3% |
| | 单词频率 | 74.2% | 74.2% | 47.4% | 46.2% | 44.5% | 43.7% |
| | 单词频率-压缩 | 65.1% | 65.1% | 43.5% | 44.7% | 41.1% | 42.6% |
| | Word2Vec | 83.0% | 83.02% | 51.0% | 49.9% | 48.7% | 47.3% |
| | CNN | 66.8% | 66.8% | 41.1% | 42.0% | 39.1% | 39.2% |
| | 统计特征 | **84.2%** | **84.2%** | **50.4%** | **51.5%** | **48.1%** | **48.3%** |
| 朴素贝叶斯 | one-hot-vector | 79.8% | 79.8% | **44.2%** | **43.8%** | **42.8%** | **42.2%** |
| | 单词频率 | **81.5%** | **81.5%** | 42.4% | 41.3% | 40.1% | 39.9% |
| | 单词频率-压缩 | 70.9% | 71.0% | 39.1% | 38.6% | 35.4% | 34.6% |
| | Word2Vec | 70.3% | 70.3% | 38.7% | 39.5% | 36.9% | 36.1% |
| | CNN | 59.0% | 59.1% | 35.7% | 37.2% | 34.2% | 34.6% |
| | 统计特征 | 81.0% | 81.0% | 39.6% | 40.3% | 39.1% | 39.3% |
| 决策树 | one-hot-vector | 65.2% | 65.2% | 34.2% | 34.0% | 32.1% | 32.4% |
| | 单词频率 | 63.9% | 63.9% | 33.2% | 32.9% | 32.3% | 32.0% |
| | 单词频率-压缩 | 60.1% | 60.1% | 32.5% | 32.6% | 31.8% | 32.1% |
| | Word2Vec | 67.9% | 66.0% | 35.8% | 35.7% | 33.7% | 33.8% |
| | CNN | 59.6% | 59.7% | 32.1% | 32.0% | 30.9% | 31.3% |
| | 统计特征 | **83.9%** | **84.0%** | **42.6%** | **44.0%** | **40.7%** | **41.3%** |
| 随机森林 | one-hot-vector | 70.2% | 70.2% | 43.5% | 43.3% | 42.6% | 42.6% |
| | 单词频率 | 69.3% | 69.3% | 42.6% | 43.0% | 41.3% | 41.7% |
| | 单词频率-压缩 | 65.5% | 65.5% | 40.7% | 40.1% | 39.8% | 39.4% |
| | Word2Vec | 79.6% | 79.6% | 45.1% | 45.2% | 44.6% | 44.4% |
| | CNN | 65.3% | 65.3% | 42.7% | 43.5% | 40.9% | 41.1% |
| | 统计特征 | **84.2%** | **84.3%** | **49.3%** | **50.9%** | **47.6%** | **48.1%** |

### 4.2　实验结果分析

对 IMDB 文本集的分类问题属于二分类问题，对 yelp2013、yelp2014 文本集的分类属于五分类问题，实验结果显示，文本情感分类的结果显示，所有分类器在二分类问题中的准确率明显高于在五分类问题中的准确率。产生该现象的原因有以下三点：a）yelp2013/14 内文本数量庞大，造成大量出现在 yelp2013/14 中的情感词没有被 MPQA 收录；b)词典 MPQA 只包含情感词，忽略了由非情感词组成的情感短语，例如："*it was on time*"，从单词层次分析，'*it*'、'*was*'、'*on*'和'*time*'单独在文本中出现时均不具有表达情感的能力，但是从短语层次分析，该短语表示"*准时*"，在客户评论中能够表达积极性观点；c)yelp2013/14 内包含大量讽刺、否定、转折句式，但是基于词袋模型创建文本向量的方法忽略了这些信息。压缩文本空间模型能够节省计算资源，缩减分类算法的运行时间。通过压缩原矩阵生成低阶近似矩阵虽然能够有效缩减文本空间模型的维度，但是在一定程度上舍弃部分信息，使得基于词频的方法创建的文本空间模型在情感分类测试中的准确率高于基于词频-压缩的方法。相比基于 CNN 的单词数据表达方法，使用 Word2Vec 模型创建的词向量在文本分类测试中具有更高的准确率。其原因是 Word2Vec 模型使用谷歌新闻作为训练语料库(大约包含 1000 亿个单词)，其规模远远大于 IMBD、yelp2013/14 (IMDB 包含 6025237 个单词、yelp2013 包含 9901789 个单词、yelp2014 包含 30087844 个单词)，使得 Word2Vec 模型创建的词向量更能够准确反映情感词的特征。并且，Word2Vec 创建的词向量包含 300 个特征，而基于 CNN 创建的词向量只包含 30 个特征，特征数量在一定程度上影响了分类器的准确率。

该实验使用词袋模型创建文本的数据表达，该方法将文本中出现的单词作为独立的个体，因此，使用该方法创建的文本向量无法记录单词之间的位置信息，进而忽略了单词间的语法依赖关系。由于语言表达存在领域依赖性(domain dependence)，即某些单词只有在特定类型的文本或者上下文环境中才具有表达情感的能力(例如：'*refund*'的意思为'*退货*'，该单词通常只在商品评论中表达消极情感)，使得文本集内存在大量单词具有表达情感的能力，但是这些单词没有被 MPQA 收录。因此，领域依赖性造成文本分类测试的情感词数量不足，降低分类算法的准确率。对比四种不同的分类器可知，SVM 更适合对文本进行情感分类，与朴素贝叶斯和决策树相比，SVM 具有更高的准确率，并且 SVM 通过计算距离实现文本分类，不必计算文本特征的先验概率和最大熵。虽然随机森林的准确率与 SVM 相当，但是随机森林的执行过程占用大量计算资源，难以实时有效的处理大规模数据。

由对照实验可知，基于统计特征创建单词和文本的数据表达的方法能够有效降低文本向量的维度，具有隐性语义空间(LSA/SVD)的压缩效果。基于统计特征创建文本向量的方法有效的减小了数据规模，降低了分类算法的复杂度，相较于基于 Word2Vec 模型的单词数据表达方法和 one-hot-vector 方法，该方法具有更高的实时性，适用对大规模文本集进行情感分类。

## 5　结束语

本文提出一种通过计算单词在文本集内的 7 中常见的分布特征，并且将七种统计特征进行组合，以低维向量的形式表示单词。实验结果显示，与其他单词的数据表达方法相比，该方法能够在保证分类算法准确率的前提下，有效的降低算法的时间和空间复杂度。下一步研究将检验该方法用在文本情感分析的其他领域中的作用例如：假新闻识别、讽刺和隐喻分析、情感强度计算。

## 参考文献：


[1] Gliozzo A, Strapparava C. Domain kernels for text categorization [C]// Proc of the 9th Conference on Computational Natural Language Learning. Stroudsburg: Association for Computational Linguistics, 2005: 56-63.

[2] May C, Ferraro F, McCree A, *et al.* Topic identification and discovery on text and speech [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2015: 2377–2387.

[3] Wang P, Qian Y, Soong F. K, *et al.* A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding [J]. arXiv: 1511. 00215v1 [cs. CL] , 2015.

[4] Šarić, F, Glavaš G, Karan M, *et al.* Takelab: Systems for measuring semantic text similarity [C]// Proc of the 1st Joint Conference on Lexical and Computational Semantics. 2012: 441-448.

[5] Iacobacci I, Pilehvar M. T, Navigli R. Sensembed: learning sense embeddings for word and relational similarity [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th

International Joint Conference on Natural Language Processing. 2015: 95-105.

[6] Kim Y. Convolutional neural networks for sentence classification [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1746–1751.

[7] Zhang H, Wang J, Zhang J, *et al.* Ynu-hpcc at semeval 2017 task 4: using a multi-channel cnn-lstm model for sentiment classification [C]// Proc of the 11th International Workshop on Semantic Evaluation. 2017: 796-801.

[8] Chetviorkin I, Loukachevitch N. Two-step model for sentiment lexicon extraction from Twitter streams [C]// Proc of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. 2014: 67-72.

[9] Pang B, Lee L, Shivakumar Vaithyanathan. Thumbs up?Sentiment classification using machine learning techniques [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2002: 79-86.

[10] Mikolov T, Chen Kai, Greg Corrado, *et al.* Efficient estimation of word representations in vector space [J]. arXiv: 1301. 3781v3 [cs. CL] , 2013.

[11] Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality [C]// Advances in Neural Information Processing Systems. 2013: 3111-3119.

[12] Mikolov T, Yih W T, Zweig G. Linguistic regularities in continuous space word representations [C]// Proc of NAACL-HLT. 2013: 746–751.

[13] Pennington J, Socher R, Manning C. Glove: global vectors for word representation [C]// Proc of Conference on Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[14] Weston J, Ratle F, Mobahi H, *et al.* Deep learning via semi-supervised embedding [M]// Neural Networks: Tricks of the Trade. Berlin: Springer, 2012. 639-655.

[15] Meng F, Lu Z, Wang M, *et al.* Encoding source language with convolutional neural network for machine translation [C]// Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015: 20-30.

[16] Segura-Bedmar I, Suárez-Paniagua V, Martínez P. Exploring word embedding for drug name recognition [C]// Proc of the 6th International Workshop on Health Text Mining and Information Analysis. 2015: 64-72.

[17] Yang J, Peng B, Wang J, *et al.* Chinese grammatical error diagnosis using single word embedding [C]// Proc of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications. 2016: 155-161.

[18] Ghosal D, Bhatnagar S, Akhtar M. S, *et al.* IITP at SemEval-2017 task 5: an ensemble of deep learning and feature based models for financial sentiment analysis [C]// Proc of the 11th International Workshop on Semantic Evaluation. 2017: 899-903.

[19] Lee Y. Y, Ke H, Huang H. H, *et al.* Combining word embedding and lexical database for semantic relatedness measurement [C]// Proc of the 25th International Conference Companion on World Wide Web. 2016: 73-74.

[20] Rajeswari K, Nakil S, Patil N, *et al.* Text categorization optimization by a hybrid approach using multiple feature selection and feature extraction methods [J]. International Journal of Engineering Research and Applications, 2014, 4 (3): 86-90.

[21] Uysal A K. An improved global feature selection scheme for text classification [J]. Expert Systems with Applications, 2016, 43: 82-92.

[22] Novoviˇcov´a Jana, Mal´ık Antonin, Pudil Pavel. Feature selection using improved mutual information for text classification [C]// Proc of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. 2004: 1010-1017.

[23] Mesleh A A. Chi square feature extraction based svms arabic language text categorization system [J]. Journal of Computer Science, 2007, 3 (6): 430-435.

[24] Mitra P, Murthy C A, Sankar K. P. Unsupervised feature selection using feature similarity [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2002, 24 (3): 301-312.

[25] Habibi M, Popescu-Belis A. Keyword extraction and clustering for document recommendation in conversations [J]. IEEE//ACM Trans on Audio, Speech, and Language Processing, 2015, 23 (4): 746-759.

[26] 韩彤晖，杨东强，马宏伟. 单词统计特性在情感词自动抽取和商品评论分类中的作用 [J/OL]. 2019, 36 (3) . [2018-02-02]. http://www. arocmag. com/article/02-2019-03-010. html.

[27] Duwairi R. M, Qarqaz I. Arabic sentiment analysis using supervised classification [C]// Proc of International Conference on Future Internet of Things and Cloud. 2014: 579-583.

[28] Lohar P, Chowdhury K. D, Afli H, *et al.* ADAPT at IJCNLP-2017 Task 4: a multinomial naive Bayes classification approach for customer feedback analysis task [C]// Proc of the 8th International Joint Conference on Natural Language Processing. 2017: 161-169.

[29] Esmaeili M, Arjomandzadeh A, Shams R, *et al.* An anti-spam system using naive Bayes method and feature selection methods [J]. International Journal of Computer Applications, 2017, 165 (4): 1-5.

[30] Kotani K, Yoshimi T. Effectiveness of linguistic and learner features to listenability measurement using a decision tree classifier [J]. Journal of Information and Systems in Education, 2016, 16 (1): 7-11.

[31] Buscaldi D, Flores J. G, Meza I. V, *et al.* SOPA: random forests regression for the semantic textual similarity task [C]// Proc of the 9th International Workshop on Semantic Evaluation. 2015: 132-133.